SPECIFICATION AMENDMENTS

Please amend the specification as follows:

Substitute the paragraph beginning on page 1, line 16 with the following:

Another method for accessing multimedia content, called a progressive download, also uses a standard Web server to supply data (e.g., a compressed media file) to a client. In this method, however, the client begins playing back the media file before the entire file is fully downloaded from the server. Thus, the time between a media selection and the beginning of playback is typically much shorter with this method than with the download-and-play method previously discussed. Playback of the media file begins during the streaming of the ~~file once~~ file, once the client has buffered a few seconds of content. The buffering provides a small backlog of information so the media can continue to play ~~uninterrupted even~~ uninterrupted, even during periods of high network congestion. With the progressive download delivery method, the client retrieves data as fast as the Web server, the network and the client will ~~allow without~~ allow, without regard to the bit-rate parameter of the compressed media stream.

Substitute the paragraph beginning on page 2, line 15 with the following:

One important aspect of accessing media content, regardless of the method of delivery, is the ability to navigate the content and/or find specific locations within the content. However, the current methods discussed above for accessing/delivering multimedia content have significant disadvantages in this regard. For example, although

some media players provide navigation functions such as fast forward and rewind, content delivery systems (e.g., Web servers, streaming media servers) may not support such accelerated or decelerated playback. Web servers, for example, are not configured to comprehend a client request for accelerated playback. In addition, even when streaming media servers support accelerated playback (or decelerated playback), the ability of a user to comprehend the content at the accelerated rate is greatly diminished because traditional streaming media servers simply drop data from media streams and only send ~~"key frame"~~ "key frames" of video to achieve the accelerated rate. Thus, there is no true acceleration ~~through~~ of the content~~, rather,~~ content. Rather, there is a "skipping" through the content. For example, a fast forward request (e.g., a request for 5 times the normal/real-time delivery/playback rate) from a client might result in the streaming media server sending only 1 video frame for every 8 seconds worth of content. This is approximately equivalent to dropping 239 out of every 240 video frames from a video stream. Thus, fast forwarding results in a jerky ~~effect as~~ effect, as if a sequence of still images is being delivered. In addition, traditional streaming media servers typically drop the entire audio stream from the media content if asked to accelerate content delivery, because the servers assume there is not enough bandwidth to send the entire stream over the network at 5 times the real-time playback rate. Also, client based media players typically drop the audio stream when fast forwarding, even when playing a local file, because they assume that the fast forwarded audio playback produces high-pitched, "chipmunk" sounding audio that is mostly incomprehensible. Furthermore, any non-continuous, non-video/audio data stream (e.g., script commands for triggering events, captions, metadata) included within the media ~~content~~ and content, and synchronized to

play at particular times during video ~~playback is~~ playback, is typically lost due to the ~~skipping through of~~ "skipping" through the video content.